

```

#include <stdio.h>
#include <stdlib.h> // nueva librería
// gcc pas_x_ref_3.c -Wall -o ejref3

void incrementar(int*);
// la función no devuelve ningún tipo de dato, solamente recibe
un INT* (puntero a entero)

int main (void)
{
    int a=1;
    printf("DIRECCION DE MEMORIA DE LA VARIABLE (en el main):
%p\n", &a);

    printf("1) Valor ANTERIOR (en el main) de 'a': %d\n", a);
    system("pause");
    incrementar(&a); // Llamo a la función incrementar() y le
paso como argumento la dirección de memoria de la variable "a"
    system("pause");
    printf("4) Valor POSTERIOR (en el main) de 'a': %d\n", a);

    return (0);
}

void incrementar(int* a)
{
    printf("DIRECCION DE MEMORIA DE LA VARIABLE (en la funcion):
%p\n", a);

    printf("2) Valor ANTERIOR (en la funcion) de 'a': %d\n", *a);
    system("pause");
    (*a) = (*a) + 1;
    printf("3) Valor POSTERIOR (en la funcion) de 'a': %d\n",
*a);
}

/*
PREGUNTAS:
Comparar los dos print de la dirección de memoria ¿que

```

diferencias encuentra? Justifique.

¿Coinciden los valores de los prints (3) y (4) en este caso?

¿Por qué?

¿Para que me podría servir utilizar funciones con punteros?

¿Que ventajas tiene el pasaje por referencia vs el pasaje por copia / valor?

*/

/* ESPACIO PARA OTRAS ANOTACIONES */